





Conhecendo o JavaScript



lá, tudo pronto para programar um pouco?

Você já aprendeu sobre várias tecnologias para desenvolver para a web até aqui. Inclusive, aprendeu também o que acontece nos bastidores até que uma página web esteja carregada, para saber se planeja o desempenho de seus jogos web. Agora vamos iniciar nossa trajetória na programação JavaScript.

Como você já sabe, a linguagem JavaScript compõem as 3 principais tecnologias para desenvolvimento web:

- HTML como uma linguagem de marcação e define o conteúdo de uma página;
- CSS que trabalha a formatação e recursos de interação e layout;
- JavaScript que irá definir o comportamento de uma aplicação web e recursos interativos complexos e dinâmicos.

Por meio do JavaScript podemos lançar mão de recursos para acessar elementos de um documento HTML, incluindo os estilos CSS. Isto é, podemos dar dinamismo aos nossos projetos inserindo novos elementos ou modificando-os dinamicamente.

Com isso, utilizaremos a linguagem JavaScript para indicar o comportamento dos nossos jogos que funcionam por meio de aplicações web. Será com ela que movimentamos objetos, detectamos teclas pressionadas, faremos

sintetização sonora, etc. Ou seja, todas as nuances de interação e experiência de jogo propriamente ditas.



Recursos-chave Fundamentais de JavaScript

O JavaScript é uma linguagem de programação dinâmica baseada em protótipos, que suporta vários paradigmas de programação. Isso quer dizer que podemos usar orientação a objetos, programação imperativa procedural ou declarativa.

Trata-se de uma linguagem leve e, por ser interpretada, é um uma linguagem de script executada no computador cliente. Isso quer dizer que podemos embuti-la dentro das tags em documentos HTML, adicionando dinamismo e complexidade em nossas aplicações e jogos.

Apesar de ser muito utilizada na web, tem sido muito popular em vários outros ambientes do lado servidor.

Sua sintaxe básica é muito similar a outras linguagens, tornando sua curva de aprendizado algo fácil. Além disso, existem uma variedade de ferramentas já desenvolvidas sobre a linguagem JavaScript para a web, o que significa que podemos acessar vários recursos extras como:

- APIs[1] já integradas aos navegadores modernos, possibilitando criar HTML e CSS em tempo de execução, acessar a webcam, usar recursos gráficos e sintetizar áudio;
- APIs de terceiros que nos permitem funcionalidades avançadas advindas de outros desenvolvedores. Empresas como o Google deixam as APIs de seus serviços públicos, como a API do YouTube por exemplo;

- Bibliotecas JavaScript de terceiros facilmente aplicáveis ao seu HTML. Facilitando a criação rápida de aplicações web. Veremos duas dessas bibliotecas úteis para nossos games nas aulas finais desta disciplina.

Como qualquer outra linguagem de programação, um programa JavaScript é composto por uma lista de instruções. Tais instruções são executadas, uma a uma, na ordem em que aparecem e são uma combinação de valores, operadores, expressões, palavras-chave e comentários.

Assim como no CSS, podemos escrever nossos programas dentro do próprio documento HTML ou em arquivos separados que serão anexados ao documento. Usaremos a tag **<script></script>** para ambos os casos. Mas não iremos nos preocupar com isso por enquanto, pois todos os exemplos que faremos nesse momento podem ser através do site jsfiddle.net.

Variáveis e Tipos de Dados

Variáveis são espaços na memória do computador onde você pode armazenar dados. Podemos declarar variáveis iniciando a instrução com as palavras-chave **var**, **let**, ou **const**, seguida pelo nome da variável definida por você. Por exemplo:

```
let nomePokemon;
```

Veja a diferença entre as três formas de declaração:

Palavra-chave	Descrição
var	Forma padrão para declarar uma variável.
let	Declara uma variável de blocos que não podem ser redeclaradas. **Como boa prática é a melhor alternativa**, pois evita erros de variáveis com o mesmo nome e consequentemente sobrescreve valores acidentalmente.
const	Variáveis declaradas assim não podem nem serem redeclaradas como seus valores não podem ser modificados. Usaremos esta declaração para os casos onde temos valores constantes.

Os nomes de variáveis em JavaScript devem ser únicos para cada variável do seu programa. São ainda *case sensitive*, isto é, diferenciam minúsculas de maiúsculas. Lembrando também que palavras-chave da linguagem não podem ser utilizadas para nomear variáveis.

Podemos atribuir valores às variáveis através do operador **=** (igual). Veja:

```
let nomePokemon = "Magnemite";
```

Em JavaScript não declaramos as variáveis com seus tipos de dados. Isso ocorre de forma dinâmica quando atribuímos valores a elas, veja alguns exemplos (adaptado de JAVASCRIPT 1, 2021):

Tipo de Dado	Explicação	Exemplo
***String* (texto)**	São valores em formato de texto. Para indicar uma string, devemos escrever o valor envolto por aspas.	let nomePokemon = "Charmander";
Número	Valores numéricos quaisquer.	let pesoPokemon = 8.5;
Booleano	Um valor que só pode ser verdadeiro ou falso. Para isso usamos as palavras-chaves **true** ou **false**.	let possuiFraqueza = true;
***Array* (vetores)**	Uma estrutura que permite armazenar vários valores em uma única variável.	Veremos mais sobre os objetos na Aula 7: Estruturas de Decisão, Repetição e Funções.
Objetos	Permite criar tipos específicos de dados que representam classes de objetos.	Veremos mais sobre os objetos na Aula 9: Indo Além do Básico com JavaScript.

Expressões e operadores

Trabalharemos com operadores representados por símbolos matemáticos, sempre com o objetivo de realizar algum cálculo computacional entre números, *strings* ou variáveis. Ou seja, podemos realizar testes e comparações, fazer expressões e cálculos matemáticos, unir sequências de texto, etc.

Vejamos a seguir, os operadores básicos e alguns exemplos em JavaScript. Lembre-se, esta lista não é definitiva, nos próximos módulos veremos outros tipos de operadores. Veremos os operadores **aritméticos** e de **atribuição**.

Os operadores de atribuição atribuem valores a variáveis JavaScript. Além do operador de atribuição de igualdade, temos outras possibilidades (adaptado de JAVASCRIP 2, 2021):

Símbolo	Utilidade	Exemplo	Expressão Equivalente
=	Atribuição de valores.	x = y	x = y
+=	Atribuição de adição.	x += y	x = x + y
-=	Atribuição de subtração.	x -= y	x = x - y
*=	Atribuição de multiplicação.	x *= y	x = x * y
/=	Atribuição de divisão.	x /= y	x = x / y

Note que alguns destes operadores são úteis para economizar linhas de código quando formos escrever operações aritméticas, que veremos a seguir.

Os operadores aritméticos realizam operações aritméticas com números literais ou variáveis. Isso será muito comum ao escrevermos os códigos de nossos jogos.

Símbolo	Utilidade	Exem
+	Adição de números e concatenação de textos.	x + y
-	Subtração de números.	x - y
*	Multiplicação de números.	x * y
/	Divisão de números.	x / y
^	Operação de expoente	x ^ y
%	Módulo da divisão.	x % y
++	Incremento de números.	x++
--	Decremento de números.	y--

Uma operação aritmética típica trabalha com dois números. Tais números podem ser literais ou variáveis.

Com números literais uma expressão exemplo pode ser:

```
let x = 10 + 21;
```

Exemplo simples de operação com variáveis:

```
let x = y + z;
```

No entanto, faremos em nossos jogos muitas operações aritméticas por meio de expressões matemáticas. Onde usaremos tanto os números literais quanto variáveis em expressões mais complexas. Por exemplo:

```
// Expressão para calcular uma nova posição no eixo x
let posX = (-1 + 5) * y - 41 / (8 + 1);
```

Podemos também usar o operador **+** também para realizar concatenação de textos:

```
let primeiroNome = "Ash";
let sobreNome = "Ketchum";
```

```
let nomePersonagem = primeiroNome + " " + sobreNome;
```

Temos ainda os operadores de comparação que serão vistos com mais detalhes no módulo Estruturas de Decisão, Repetição e Funções.

Chegamos ao final de mais um resumo. No próximo encontro vamos avançar na programação com JavaScript, aprendendo sobre novos operadores e como trabalhar com estruturas de dados mais complexas.

Abraços!

[1] Uma API (*Application Programming Interface*) pode ser definida como um conjunto de código e regras prontas para que outros programadores possam criar softwares que se comunicam entre si. Assim podemos criar softwares que acessam componentes de software de terceiros através da internet.



Atividade Extra

O Reino encantado do JavaScript

Como a nossa unidade tem como objetivo apresentar a linguagem JavaScript, que tal conhecer um pouco mais, principalmente sobre a evolução da linguagem? Acesse o podcast abaixo para conhecer a história do JavaScript, como ele evoluiu para se tornar uma ferramenta poderosa para construirmos aplicações web e jogos com interatividade e experiências poderosas. Será importante que você conheça essas nuances da linguagem, para sempre projetar adequadamente seus jogos e aplicações.

- **O Reino encantado do JavaScript – Hipsters #38.** Disponível em: <https://hipsters.tech/o-reino-encantado-do-javascript-hipsters-38/>.

Todo mundo ama matemática!

A comunidade de desenvolvedores da Mozilla criou um material muito interessante sobre matemática básica, números e operadores no JavaScript. Acesse o conteúdo dado pelo link abaixo, leia e execute os experimentos que são pedidos neste material no console do navegador. Isso será um bom exercício, pois em jogos sempre será necessário trabalhar corretamente com números e cálculos matemáticos.

- **Matemática básica no JavaScript – números e operadores.** Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/Math>.



Referência Bibliográfica

BONATTI, D. **Desenvolvimento de Jogos em HTML5.** Rio de Janeiro: Brasport, 2014.

MDN WEB DOCS. **JavaScript básico.** Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/JavaScript_basics>.

W3 Schools. **JavaScript Tutorial.** Disponível em: <https://www.w3schools.com/js/default.asp>.

Ir para exercício